

# Algoritmos de planificación en redes de paquetes

## *Scheduling algorithms in packet networks*

Juan Carlos Cuéllar Quiñónez, Esp.  
Universidad Icesi – Colombia  
[jcuellar@icesi.edu.co](mailto:jcuellar@icesi.edu.co)

Fecha de recepción: 25-03-2009

Fecha de selección: 28-10-2009

Fecha de aceptación: 03-08-2009

### ABSTRACT

This article presents the mechanisms and processes required in order for a router to provide QoS in packet networks and it presents the state of the art in the subject of scheduling algorithms use in this kind of networks.

### KEYWORDS

Quality of Service, Congestion Management, Congestion Avoidance, Admission Control, RED, GPS, WFQ, DRR, PQ, CQ.

### RESUMEN

En este artículo se presentan los mecanismos y procesos necesarios

que realizan los enrutadores para ofrecer calidad de servicio en redes de paquetes y se hace especial énfasis en el estudio del estado del arte en lo referente a los algoritmos de planificación utilizados en este tipo de redes.

### PALABRAS CLAVE

Calidad de servicio, administración de la congestión, evasión de la congestión, control de admisión, RED, GPS, WFQ, DRR, PQ, CQ.

**Clasificación Colciencias:** Tipo 3.

## I. INTRODUCCIÓN

En sus inicios Internet se diseñó para proporcionar conectividad entre sitios remotos, y las aplicaciones y servicios que se utilizaban no requerían condiciones especiales en la red para su funcionamiento, ya que los datos viajaban basados en el concepto de mejor esfuerzo (*best-effort*), la red transportaba los datos, pero no garantizaba calidad de servicio. Adicionalmente, aplicaciones diferentes a los datos, como voz o video, tenían sus propias redes independientes y paralelas a la red de datos. La utilización de este esquema, con distintas plataformas o tipos de red, una por cada servicio o aplicación, hacía de la administración de la red un proceso complejo y costoso.<sup>1</sup>

Con el avance tecnológico fueron apareciendo nuevas aplicaciones y servicios (video multimedia, Voz sobre IP, video streaming, etc) que requerían condiciones de calidad de servicio especiales para funcionar de manera adecuada y mantener satisfecho al usuario final. Fue entonces cuando surgió el concepto convergencia, que pretende integrar las aplicaciones y los servicios en una única plataforma de red. Fue así como se produjo una evolución de las redes tradicionales a un nuevo tipo de red con una arquitectura única, éstas fueron llamadas redes de próxima generación (*Next Generation Networks – NGN*).<sup>1,33,34</sup>

Ahora bien, uno de los aspectos primordiales en este tipo de redes es garantizar calidad de atención extremo a extremo a las aplicaciones o servicios, de acuerdo con sus requerimientos, para esto, diferentes organismos de estandarización han propuesto arquitecturas para garan-

tizar la calidad del servicio (*QoS*) en este tipo de redes.

Cada organismo plantea una arquitectura según el sector de la tecnología en que esté inmerso; por ejemplo Cable Labs,<sup>35</sup> y DSL Forum<sup>36,48</sup> definen una arquitectura para control de calidad de servicio para la red de acceso en la que se especializan redes de acceso de HFC y DSL respectivamente, mientras que la ITU-T y la ETSI definen una arquitectura genérica que cubre otros tipos de tecnologías.<sup>2</sup> Las tecnologías inmersas en el núcleo (*core*) de las redes (*MPLS*<sup>37,47</sup> y *Carrier Ethernet*)<sup>38</sup> ofrecen también mecanismos de calidad de servicio de acuerdo con sus procesos de funcionamiento.

La IETF propuso dos mecanismos para garantizar calidad de servicio: Servicios Integrados (*Integrated Services –IntServ-*)<sup>3</sup> y Servicios Diferenciados (*Differentiated Services – DiffServ-*).<sup>4,46,47</sup> pero con la evolución en las aplicaciones y servicios convergentes, estos mecanismos no garantizaban una calidad que permitiera que aplicaciones y servicios funcionaran de manera óptima. Esto llevó a que diferentes organismos y fabricantes propusieran mejoras a DiffServ e IntServ convirtiendo a estos dos mecanismos en la base de muchas arquitecturas que ofrecen calidad de servicio.

Con base en lo anterior, es claro que para que las aplicaciones/servicios en este tipo de redes convergentes funcionen de manera adecuada y brinden satisfacción al usuario, se les debe garantizar una calidad de servicio para su funcionamiento.<sup>51</sup>

Para garantizar calidad de servicio extremo a extremo en una red con-

vergente se deben tener en cuenta los siguientes aspectos:<sup>5</sup>

- Configuración de procedimientos de calidad de servicio en los dispositivos de red, en los cuales se deben realizar labores de encolamiento (*queueing*), planificación (*scheduling*) y conformación del tráfico (*traffic shaping*).
- Técnicas de señalización entre los diferentes dispositivos de red para garantizar calidad de servicio extremo a extremo.
- Funciones de administración y control del tráfico que va a transitar a través de la red.

Lo anterior se realiza en los enrutadores, que son los encargados del envío de los paquetes: estos enrutadores se ubican ya sea en los límites o borde de la red o en la espina dorsal (*backbone*) de la misma. Dependiendo de la ubicación del enrutador, éste realiza unas tareas específicas para garantizar la calidad de servicio.

Los enrutadores de borde se encargan de las siguientes funciones:

- Clasificar los paquetes
- Control de Admisión (*Admission control*)

- Administración de la congestión (*congestion management*)

Los enrutadores de *backbone* se encargan de las siguientes funciones:

- Administración de la congestión (*congestion de management*)
- Evasión de la congestión (*congestion de avoidance*)

En la Figura 1 se pueden apreciar de manera resumida las tareas que debe llevar a cabo un enrutador para garantizar la calidad del servicio.

El objetivo del artículo es analizar cada uno de estos procesos, con especial énfasis en los algoritmos de planificación, revisar el estado del arte; y mediante simulaciones, establecer parámetros de comparación entre varios de ellos.

El artículo está organizado de la siguiente manera: en la sección 2 se hace una breve descripción de los mecanismos de evasión de la congestión, en la sección 3 se describe el proceso de clasificación, aplicación de políticas (*policing*) y conformación del tráfico (*shaping*), en la sección 4 se detallan los principales mecanismos de administración de la congestión

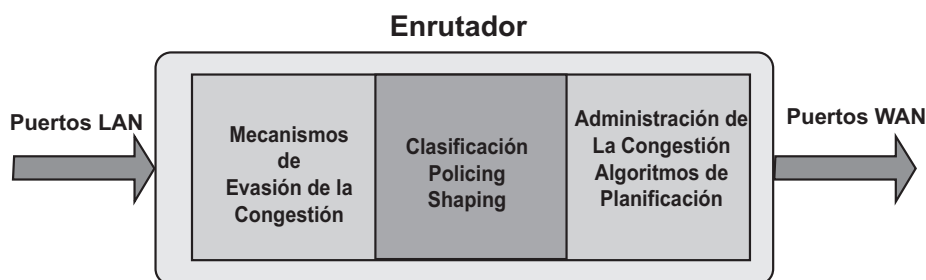


Figura 1. Procesos para manejar QoS en un enrutador.

que están relacionados con los algoritmos de planificación y al finalizar se presentan las conclusiones y el trabajo posterior.

## 2. MECANISMOS PARA LA EVASIÓN DE LA CONGESTIÓN

Los mecanismos para la evasión de la congestión (*congestion avoidance*) son los encargados de monitorear el tráfico que llega al enrutador y anticiparse a la congestión, entendiéndose que hay congestión cuando la demanda por un recurso (por ejemplo, ancho de banda de un enlace) excede la capacidad disponible del recurso. El resultado de esto son tiempos de espera y respuesta elevados en las aplicaciones, y pérdida de paquetes.<sup>6</sup>

El mecanismo tradicional que utilizan los enrutadores para la evasión de la congestión es el descarte en la cola (*Tail Drop*) que consiste en descartar paquetes cuando la cola se ha llenado, esto tiene el inconveniente que bloquea las colas y hace que las colas se llenen de paquetes, es decir, se le informa a las estaciones que hay congestión solo cuando la cola está llena.

Para solucionar este inconveniente se planteó el mecanismo de administración activa de la cola (*Active Queue Management - AQM*), que consiste en descartar paquetes antes que la cola se llene, e informar al equipo fuente para evitar así la pérdida excesiva de paquetes. Un ejemplo de AQM es *Random Early Detection (RED)* que se ha convertido en un estándar *de facto* con múltiples variaciones.

RED fue propuesto por Sally Floyd y Van Jacobson en los años noventa<sup>7</sup> y es la base de todos los algoritmos

AQM. Su funcionamiento radica en controlar el tamaño de la cola de recibo de paquetes e indicar a las estaciones cuándo disminuir la tasa de envío de paquetes de manera temporal.<sup>5</sup> Se apoya en el mecanismo de control de congestión utilizado por TCP, hace descarte de paquetes de manera aleatoria en periodos de congestión y le indica a la estación generadora de paquetes que disminuya su tasa de transmisión, mientras los paquetes alcanzan su destino y se revela que la congestión ha disminuido.

Actualmente hay variaciones del RED<sup>8</sup> original, como por ejemplo *Adaptive RED (ARED)*,<sup>9,50</sup> *Dynamic-RED (DRED)*,<sup>10,49</sup> *Stabilized-RED (SRED)*,<sup>11</sup> *BLUE*,<sup>12</sup> y *Adaptive Virtual Queue (AVQ)*.<sup>13</sup>

Por otro lado, Cisco Systems implementó *Weighted Random Early Detection (WRED)*, que combina las características de RED y el campo *IP Precedence*<sup>51</sup> del datagrama IP, y descartar los paquetes con menor valor en el campo IP precedence antes que los paquetes con mayor valor en este campo. Adicionalmente, dicha empresa ha implementado también *Distributed WRED (DWRED)* y *DiffServ Compliant WRED*.<sup>5</sup>

Continúa el estudio que busca mejorar los algoritmos AQM, y se ha presentado *ProActive Queue (PAQMAN)*,<sup>14</sup> el cual captura el tráfico entrante y regula el tamaño de la cola de acuerdo con un nivel de servicio deseado.

## 3. CLASIFICACIÓN, POLICING Y SHAPING

### A. Clasificación

La clasificación de paquetes permite identificar y dividir el tráfico de la

red en múltiples niveles de prioridad o clases de servicio. Esta clasificación se puede realizar de muchas maneras, al usar atributos del paquete tales como *Differentiated Services Code Point (DSCP)*, *IP Precedence*, dirección IP fuente o dirección IP destino.<sup>15</sup>

Por ejemplo, cuando se usan los tres bits en el campo de tipo de servicio (ToS) en el encabezado del paquete IP, dichos paquetes se pueden clasificar hasta en seis clases (dos de los valores son usados para otros propósitos). Adicionalmente los paquetes también pueden ser clasificados por fuentes externas, ya sea por los equipos del cliente o por los equipos del proveedor de servicios de interconexión.

Otro de los mecanismos para el marcado y clasificación de paquetes es *Differentiated Services Code Point (DSCP)*, que asigna prioridad a los paquetes de acuerdo al campo de 6 bits que se posee en el encabezado IP, por lo tanto, según el tipo de tráfico hay una clase asignada para él, con los requerimientos que demande para su buen funcionamiento. El inconveniente de este mecanismo radica en que se requiere de mucho procesamiento en los enrutadores para marcar y clasificar los paquetes, como también lo correspondiente a la señalización, que carga la red. Otro problema latente consiste que en aplicaciones de características similares, pero de diferente importancia para la organización, pueden ser clasificadas en el mismo tipo y recibir igual prioridad, lo cual hace que compitan por el ancho de banda disponible en el momento de ser transmitidas.<sup>16</sup>

## B. Aplicación de políticas (*Policing*)

El *Policing* es el mecanismo por el cual se descartan paquetes cuando se han definido unos límites o políticas de funcionamiento. Este mecanismo permite descartar los paquetes que tengan menor prioridad en su envío y se puede usar en interfaces tanto de entrada como de salida de los enrutadores.

Un ejemplo del uso de *policing* se ve reflejado cuando un usuario empieza a utilizar mayor ancho de banda del contratado con su proveedor. El *policing* hace que el usuario disminuya la rata de transmisión de paquetes.<sup>15</sup>

## C. Conformación (*Shaping*)

El *Shaping* se emplea en las interfaces de salida de los enrutadores, es decir, en interfaces que se conectan al *core* de la red del proveedor de servicios, y consiste en garantizar que el tráfico que proviene de una interfaz de alta velocidad no sature una interfaz de baja velocidad, es decir, el enrutador se encarga de modelar el tráfico de tal manera que no haya pérdida de paquetes al emplear el enlace de salida. Esto es muy utilizado por los proveedores de servicio y los usuarios para garantizar siempre el ancho de banda contratado.<sup>15</sup>

## 4. ALGORITMOS DE PLANIFICACIÓN

Como se ha revisado en las secciones anteriores, para ofrecer calidad de servicio en una red convergente es necesario realizar una serie de procesos de manera conjunta y sincronizada.

En esta sección se abordará lo referente a los algoritmos de planifica-

ción, los cuales tienen que ver con la rutina que debe ejecutar el enrutador para ir despachando los paquetes por uno de sus puertos de salida, lo cual depende de la prioridad que se le haya asignado a cada paquete. Esta prioridad depende de la aplicación o servicio que esté transportando el paquete.

Estos algoritmos de planificación son parte fundamental del proceso que debe realizar el enrutador en lo correspondiente a la administración de la congestión, o es el proceso mediante el cual se crean las colas y se asignan paquetes a dichas colas, lo que depende de la clasificación o marcación realizada a cada paquete. Después de esto se encarga de definir la manera de despachar los paquetes sobre la interfaz de salida, esta tarea la realiza el algoritmo de planificación.

Uno de los algoritmos de planificación (*packet scheduler*) más simples es el FCFS (*First Come First Served*), que despacha los paquetes de acuerdo con su orden de llegada. Este algoritmo es fácil de implementar pero no ofrece ningún tipo de QoS ya que cada flujo de información toma una porción de ancho de banda de manera arbitraria.<sup>17</sup>

De manera general, un algoritmo de planificación debe tener las siguientes características:<sup>17</sup>

1. **Ser equitativo en la asignación de recursos:** El algoritmo debe proveer aislamiento entre los múltiples flujos con comparten el enlace, y en particular cada flujo debe tener disponible un ancho de banda determinado que no afecte su propio funcionamiento, y el de otros flujos.

## 2. Limitación en el retardo del paquete:

Aplicaciones como video o telefonía IP requieren de un retardo extremo a extremo que esté dentro de unos límites que permitan que el usuario quede satisfecho con el funcionamiento de la aplicación. Por lo tanto, el algoritmo de planificación debe definir el orden en el cual los paquetes van a ser enviados y así determinar el tiempo de retraso dentro de la cola en cada enrutador, para lograr obtener un retraso total del paquete, de manera que no se afecte el funcionamiento y desempeño de las aplicaciones.

3. **Baja complejidad:** Con el incremento de la velocidad de los puertos LAN de un enrutador, se hace crítico el tiempo que demora el algoritmo de planificación analizando cada paquete, esta operación es del orden de los nanosegundos, así que si el algoritmo de planificación es muy complejo hará que este tiempo de análisis aumente, como también la complejidad en el hardware para que pueda desempeñarse esta operación.

Diseñar un algoritmo de planificación que cumpla con las características enunciadas arriba se convierte, por lo tanto, en una tarea compleja. El estudio y diseño de estos algoritmos permiten identificar claramente dos enfoques: algoritmos basados en el concepto de marcas de tiempo (*timestamp*) y algoritmos basados en la estrategia *round-robin*.

En las siguientes secciones se explicará el funcionamiento básico de cada enfoque y algunas variaciones diseñadas basándose en cada enfoque.



### A. Algoritmos de planificación basados en *Timestamp*

En los primeros estudios que se realizaron sobre este tema, Generalized Processor Sharing (GPS),<sup>18,42</sup> es considerado como el algoritmo de planificación ideal que alcanza un perfecto aislamiento entre los flujos y una asignación de ancho de banda equitativo a los mismos, pero este algoritmo es de muy difícil implementación. Su funcionamiento se ha convertido en punto de referencia para los algoritmos basados en *timestamp*.<sup>17</sup>

Estos algoritmos basados en *timestamp* emulan el funcionamiento y operación de GPS calculando un *timestamp* para cada paquete. El algoritmo de planificación típico con el cual se puede explicar este funcionamiento es el *Weighted Fair Queueing (WFQ)*.<sup>39,40,41</sup>

En WFQ, los paquetes son enviados en el orden de sus tiempos de finalización (*finish times*). Para calcular dichos tiempos WFQ almacena dos variables (asumiendo que los pesos para todas las colas son iguales a "1"):

- Número de turno actual.
- Tiempo de finalización (información del estado del paquete almacenada por cola)

El número de turno actual es el número asignado al servicio de cada cola y refleja la cantidad de tráfico a enviar en cada cola.<sup>19</sup>

Para calcular el tiempo final ( $F_i$ ) en cada cola  $i$ , se supone un paquete de longitud  $L$ , que llega cuando la cola está vacía y tiene un número de turno  $R$ . El tiempo final para cada

paquete corresponde entonces al tiempo de transmisión del último bit del paquete ( $F_i = R + L$ ). El paquete será enviado por el algoritmo WFQ únicamente cuando el tiempo de finalización sea menor en comparación con los tiempos calculados en las otras colas. Por esta razón, el WFQ da mayor prioridad a flujos de tráfico cuyos paquetes sean más pequeños, en relación con los de otros flujos.

Cuando se asignan pesos a las colas este peso entra a ser parte de la expresión para calcular el tiempo final, por lo tanto una asignación de peso alta conlleva un tiempo de finalización pequeño y por ende mayor ancho de banda para el tráfico del flujo.

En la Figura 2 se puede apreciar la comparación entre FCFS y WFQ para tres colas, se ve cómo cambia el orden de envío de los paquetes en WFQ (los paquetes de menor tamaño son enviados mucho más rápido).

Se puede resumir la operación de WFQ en cuatro pasos:<sup>19</sup>

- Cuando el paquete llega, es clasificado y ubicado en la cola respectiva.
- El número de turno se recalcula.
- El tiempo de finalización del paquete se calcula
- Se envía el paquete con el menor tiempo de finalización.

En resumen, se puede decir que este tipo de algoritmos de planificación son buenos en lo referente al retraso total del paquete, pero su implementación es muy compleja, lo que hace que el costo computacional sea alto.

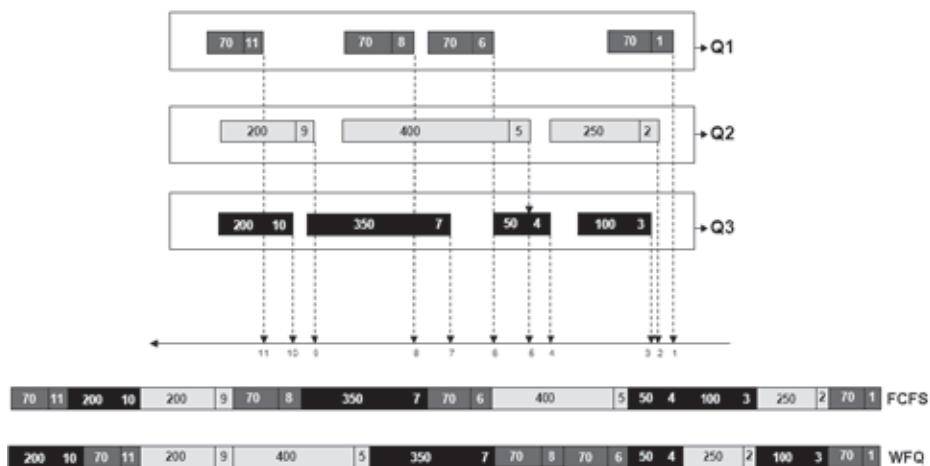


Figura 2. Comparación entre FCFS y WFQ.<sup>28</sup>

### Tipos de algoritmos WFQ

Hay una gran variedad de algoritmos WFQ. Se explicará a continuación el funcionamiento de los utilizados por la compañía Cisco en sus enrutadores : WFQ, DWFQ, CBWFQ y DCBWFQ .

#### Flow-based WFQ (WFQ)

Su funcionamiento es similar al WFQ explicado en la sección anterior, donde al tráfico de menor volumen se le asigna mayor prioridad de envío que el tráfico de mayor volumen. WFQ clasifica los flujos basándose en el encabezado del paquete, donde puede analizar la dirección IP o dirección MAC fuente o destino, puertos origen o destino o valor ToS (Tipo de Servicio).

WFQ no es soportado en interfaces túnel y con cifrado, debido a que estas características modifican la información del contenido del paquete utilizada para su clasificación. Este algoritmo puede trabajar con *IP precedence* y permite detectar la

prioridad asignada al paquete proporcionando una respuesta más rápida en su envío.<sup>5</sup>

#### Distributed Weighted Fair Queuing (DWFQ)

DWFQ es una versión especial de alta velocidad de WFQ que corre sobre interfaces VIP (Versatile Interface Processor) en enrutadores de las series 7000 y 7500. Existen dos tipos de DWFQ:

- **Basado en flujo (*Flow-based*):** Los paquetes son clasificados por flujo, identificando cada flujo por la misma dirección IP fuente o destino, puertos fuente o destino TCP o UDP y el campo de ToS. Los flujos son tratados de igual manera.
- **Basado en clase (*Class-based*):** Los paquetes son asignados en colas diferentes según su grupo de QoS o del *IP precedence* asignado en el campo ToS. Mediante estos grupos de QoS se puede personalizar las políticas de QoS en cada flujo.<sup>5</sup>



### Class-Based Weighted Fair Queueing (CBWFQ)

CBWFQ permite definir clases de tráfico por usuario, y al administrador establecer un criterio para identificar el tráfico que depende de la clase a que éste pertenezca, y así asignar ancho de banda, peso y máxima longitud de transmisión del paquete. El ancho de banda asignado a una clase es garantizado durante periodos de congestión.<sup>5</sup>

La suma de todos los anchos de banda de los flujos que van a salir por una interfaz no debe superar el 75% del ancho de banda disponible en dicha interfaz, el 25% restante es usado para información de control como tráfico de enrutamiento, encabezados de niveles 2 y 3.

### Distributed Class-Based Weighted Fair Queueing (DCBWFQ)

DCBWFQ soporta diferentes clases de tráfico por usuario ejecutándose en la VIP (Versatile Interface Processor), permite configurar el tamaño de la cola y las políticas de servicio al

utilizar listas de acceso que permite mayor desempeño en el momento de enviar los paquetes de cada cola.<sup>5</sup>

En la Tabla 1 se puede apreciar un cuadro comparativo entre los cuatro tipos de versiones de WFQ analizadas.

Adicionalmente hay otros algoritmos basados en *timestamp*, entre los cuales están: *Virtual clock*,<sup>20</sup> Packetized Generalized Processor Sharing (PGPS),<sup>21</sup> Self-clocked fair queueing (SCFQ),<sup>22</sup> y Fair Weighted fair queueing (WF<sup>2</sup>Q),<sup>23,41</sup>

### B. Algoritmos de planificación basados en Round-Robin

Los algoritmos de planificación basados en Round-Robin,<sup>24,25,26</sup> asignan una ranura de tiempo (*time slot*) a cada flujo, y el envío de paquetes se hace de manera alternada, flujo por flujo, en una secuencia. Por ejemplo, si se tuvieran 3 colas (Q1, Q2 y Q3) a cada una se le asignaría una ranura de tiempo de duración  $t_1$ . El algoritmo empezaría a enviar los paquetes de Q1 durante un tiempo

**Tabla 1** Comparación entre WFQ, DWFQ, CBWFQ y DCBWFQ<sup>5</sup>

WFQ	DWFQ	CBWFQ	DCBWFQ
Flow-based WFQ: - Asigna pesos al clasificar los paquetes. - Cuando los paquetes no son clasificados se distribuye de manera equitativa el ancho de banda.	Flow-based DWFQ: - Distribución equitativa del ancho de banda sin asignación de pesos. - Class-based DWFQ: - Asignación de pesos. - Grupos de QoS. - Tipo de Servicio.	Class-based WFQ: - Asignación de pesos. - Asignación de ancho de banda dependiendo de la clase de tráfico.	Class-based distribute +-d WFQ: - Asignación de pesos. - Asignación del ancho de banda dependiendo de la clase de tráfico.
Corre en plataformas estándar Cisco IOS	Corre en VIP (Versatile Interface Processor) permitiendo mayor desempeño	Corre en plataformas estándar Cisco IOS	Corre en VIP (Versatile Interface Processor) permitiendo mayor desempeño

t de duración  $t_1$ , después enviaría paquetes de Q2 en un tiempo  $t_1$  y posteriormente enviaría paquetes de Q3 en  $t_1$ , para nuevamente retornar a la cola Q1 para el envío de paquetes.

Este tipo de algoritmo es mucho más fácil de implementar que WFQ, pero tiene una respuesta muy pobre con respecto al retraso obtenido para cada paquete. Se documentará el funcionamiento de los principales algoritmos que funcionan con este esquema para posteriormente, en la siguiente sección, estudiar los algoritmos híbridos que permiten mejorar las características de desempeño al utilizar componentes de los algoritmos basados en timestamp y round-robin.<sup>17</sup>

### Weighted Round Robin (WRR)

Este algoritmo es una modificación del round robin, en el cual solo se envía un paquete por cola en cada turno de servicio. A cada flujo se le asigna un peso que corresponde a un porcentaje del ancho de banda del enlace que utilizará. El número de paquetes a enviar en cada turno se calcula con base en el peso del flujo y en la capacidad del enlace.

WRR funciona bien para el envío de paquetes de tamaño fijo, pero presenta problemas de desempeño cuando los paquetes son de longitud variable, ya que en el caso de paquetes de tamaño mayor utiliza más ancho de banda del asignado, lo cual hace que el desempeño en el envío de paquetes de las otras colas disminuya.<sup>27,43</sup>

Este esquema se emplea en dispositivos de interconectividad Huawei,

en los cuales se manejan ocho colas a las cuales se les puede asignar un porcentaje de ancho de banda determinado para garantizar QoS.<sup>32</sup>

### Deficit Round Robin (DRR)

Este algoritmo mejora los inconvenientes que presenta el WRR y permite así enviar paquetes de longitud variable con un mejor desempeño. Funciona en dos variables, una llamada *quantum* que indica la cantidad de bits a enviar en cada turno de transmisión y otra llamada contador de déficit (*deficit counter*) que se encarga de almacenar el valor del *quantum* obtenido al transmitir un paquete y que permitirá posteriormente enviar paquetes de longitud variable.

El algoritmo empieza su funcionamiento enviando paquetes de las colas que tengan paquetes por mandar. Si el tamaño del paquete es menor o igual al valor del *quantum*, se envía. Sin embargo, si la longitud del paquete es mayor que el valor del *quantum*, el paquete debe esperar a ser transmitido en otro turno. En este caso el *deficit counter* se inicializa para esta cola y si el paquete no puede ser enviado en un turno, el *deficit counter* es incrementado en el valor que posea el *quantum*. Cuando el *deficit counter* es igual o mayor al tamaño del paquete (en el caso de paquetes mayores que el valor del *quantum*) se envía el paquete y el *deficit counter* vuelve a cero.<sup>27</sup>

En la Figura 3 se puede apreciar la comparación entre una cola que trabaja con FCFS y otra que lo hace con el esquema de DRR.

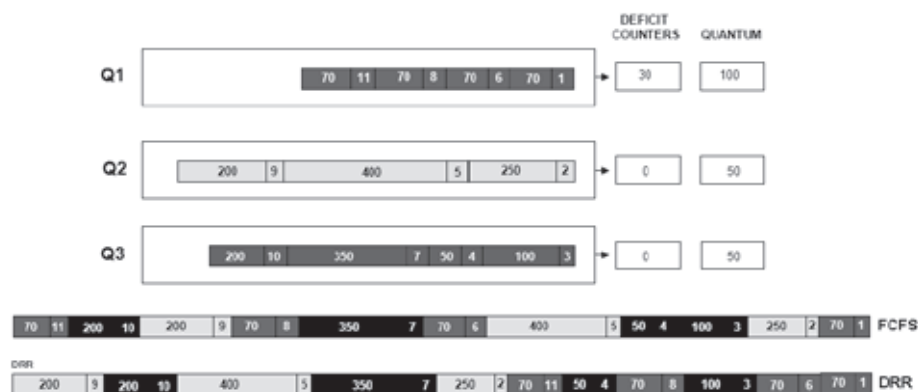


Figura 3. Comparación entre FCFS y DRR.<sup>28</sup>

### Custom Queueing (CQ)

Este tipo de algoritmo trabaja con 17 colas; la cola 0 es utilizada para llevar información del sistema y las colas de la 1 a la 16 se pueden asignar a tráfico de los usuarios.

CQ funciona enviando paquetes mediante una atención round robin de manera secuencial, atendiendo todas las colas configuradas turno por turno. El algoritmo permite definir la cantidad de bytes a enviar en cada turno, y con esto se puede asignar mayor ancho de banda a una cola determinada.<sup>5</sup>

Se debe ser muy cuidadoso en el cálculo de los bytes a enviar, ya que dicha cantidad se relaciona con el tamaño de paquete que maneje el protocolo que utiliza la cola. Adicionalmente se configura una cola por omisión en la cual se almacena y envía todo el tráfico que no ha sido tenido en cuenta en las otras colas que se han configurado.

### Priority Queueing (PQ)

Este esquema funciona con cuatro colas cuyas prioridades son alta, me-

dia, normal y baja. La asignación de paquetes a cada cola depende de las características de funcionamiento de cada aplicación. El orden de atención de las colas depende de la prioridad de la misma, siendo la más prioritaria la cola alta, luego la media y así sucesivamente.<sup>5</sup>

Cuando se configura este tipo de algoritmo de planificación se deben analizar de manera detallada las aplicaciones a asignar a cada cola, porque el algoritmo funciona enviando primero paquetes de la cola de prioridad alta y cuando esta está vacía pasa a enviar paquetes de la cola media, y así sucesivamente, hasta llegar al envío de paquetes de la cola de prioridad baja, pero ocurre que si se están enviando paquetes, por ejemplo, de la cola de prioridad normal y llegan paquetes a la cola de prioridad alta, el envío de paquetes en la cola de prioridad normal se detiene y se empiezan a enviar paquetes de la cola de prioridad alta hasta que esta esté vacía.

Hay otras variantes de algoritmos basados en DRR como DRR+<sup>26</sup> y

DRR++,<sup>29</sup> estos dan mayor prioridad a una o a varias colas para minimizar la latencia en el despacho del paquete.

Otra variante es el Modified Deficit Round Robin (MDRR), que se implementa en enrutadores comerciales. En el caso de la empresa Cisco, es implementado en enrutadores de la serie 12000 bajo el nombre de C-MDRR,<sup>30</sup> en el cual se define una cola de baja latencia, que tiene mayor prioridad sobre las otras y se utiliza para el tráfico de voz. En los dispositivos M-Series de Juniper el algoritmo es llamado J-MDRR,<sup>31</sup> y define tres colas de prioridad baja, alta y estrictamente alta.

### C. Algoritmos híbridos

Los algoritmos de planificación estudiados permiten enviar paquetes con una prioridad estricta. WFQ y DRR pueden garantizar ancho de banda y ciertos límites de retraso en cada cola, sin embargo no garantizan baja latencia en términos generales.

A raíz de esta serie de inconvenientes, se han desarrollado algoritmos híbridos que lo que hacen es adaptar en uno solo las características de los algoritmos WFQ y DRR, para mejorar el desempeño y minimizar los retrasos en el envío de paquetes.

Un algoritmo híbrido funciona de la siguiente manera: Los flujos son agrupados en  $N$  clases, por lo tanto los flujos de la misma clase reciben una proporción similar de ancho de banda. El algoritmo tiene dos estados: Selección de clases y selección de flujo. Para cada turno a enviar, el algoritmo primero selecciona la clase y después selecciona el flujo que está dentro de la clase. El *timestamp* se

aprovecha para planificar entre las clases y el *round robin* para planificar dentro de las clases.

Un ejemplo de algoritmo híbrido es el *Low Latency Queueing (LLQ)*, también conocido como PQ/CBWFQ, en el cual se combinan las características de PQ para clasificar el tráfico y definir las colas; el envío de paquetes se realiza basándose en CBWFQ. Este tipo de esquema divide el tráfico en cuatro clases: uno de clase prioridad alta, dos clases de ancho de banda garantizado y una clase para tráfico predeterminado. El tráfico de prioridad alta es asignado a una cola de prioridad, el tráfico con ancho de banda garantizado es ubicado en colas reservadas y el tráfico predeterminado se ubica en una cola sin reserva. El tráfico de prioridad alta es despachado en el primer turno, a menos que se exceda el ancho de banda que le fue asignado, las colas de reserva son atendidas de acuerdo con el ancho de banda asignado, dicho ancho de banda es usado para calcular un peso, que se usa para determinar qué tan a menudo la cola de reserva se puede atender y cuántos bytes puede enviar por turno. Un aspecto importante de este algoritmo es que durante periodos de congestión no monopoliza el ancho de banda disponible para una sola cola y se recomienda para VoIP.

Entre otros algoritmos híbridos están: *Group Round-Robin (GRR)*,<sup>52</sup> *Fair Round Robin (FRR)*,<sup>53</sup> y *Stratified Round Robin (StratRR)*.<sup>54</sup>

### 5. CONCLUSIONES

Ofrecer QoS en una red convergente es un proceso complejo, que implica una serie de tareas que son desco-

nocidas para el usuario final, pero que si no se realizan de manera adecuada, hace que el servicio prestado no cumpla parámetros mínimos de funcionamiento.

Actualmente, las técnicas empleadas para ofrecer calidad de servicio se utilizan casi que con una configuración por omisión (o de fábrica) en los diferentes dispositivos de interconectividad que los fabricantes ofrecen a los ISP o proveedores de servicio, y esto se debe a que los anchos de banda de los enlaces en sus redes de *core* son suficientes y no presentan congestión alguna.

Pero con la aparición de nuevos servicios, por ejemplo Televisión sobre IP (IPTV), que utilicen estos enlaces, esta situación puede cambiar al hacer que los enlaces se congestionen y provoquen degradación en el desempeño de las aplicaciones. Es entonces cuando se debe elegir un algoritmo de planificación o un esquema de encolamiento adecuado que permita mantener los parámetros de funcionamiento correctos en las aplicaciones que utilice el usuario final.

Para poder elegir un algoritmo de planificación o esquema de encolamiento apropiado se deben analizar múltiples factores. Uno de ellos es la capacidad de los enrutadores; obviamente entre más robusto sea el enrutador de borde mucha más capacidad en hardware y software se podrá tener, y por ende muchas más prestaciones. Estas prestaciones se ven reflejadas en el tiempo que permanecen los paquetes en la cola. Se debe realizar adicionalmente un análisis de tráfico, e identificar las aplicaciones críticas para el funcionamiento de la organización, y con base

en esto establecer qué prioridad se le debe asignar a cada aplicación para su envío. Paralelo a esto se debe elegir el algoritmo de planificación que garantice su buen funcionamiento al permitir que el usuario esté satisfecho, sin degradar el funcionamiento de otras aplicaciones menos críticas o que se les ha asignado menor prioridad.

El ofrecimiento de QoS en una red convergente es un proceso complejo que involucra muchas variables. Al ser tan complejo, los administradores de red se pueden apoyar en herramientas de simulación para analizar qué puede pasar con sus enlaces ante diferentes tipos de tráfico, y así poder realizar los ajustes necesarios en los dispositivos de interconectividad que estén funcionando, sin que el usuario final note este tipo de labores.

## 6. TRABAJO POSTERIOR

Este artículo es la base para el trabajo que sigue en el proyecto de grado de maestría titulado *Caracterización y comparación entre los mecanismos de control de calidad de servicio utilizados en redes de próxima generación*, en el cual se realizará la simulación de clientes xDSL generando tráfico con diferentes servicios como datos, voz y video, y utilizando como core una red MPLS. La idea es analizar qué tipo de algoritmo de planificación entrega mejores resultados en cuanto al desempeño para este tipo de aplicaciones y con base en estos tener las herramientas suficientes para configurar determinado esquema de encolamiento cuando se presente congestión en la red. Esto podrá ocurrir cuando se empieza a ofrecer IPTV en las redes de los diferentes operadores o ISP de la región.

## 7. BIBLIOGRAFÍA

- 1 Knightson, K. Morita, N., Towle T. *NGN Architecture: Generic Principles, Funcional Architecture, and Implementation*. IEEE Communications Magazine. Octubre 2005.
- 2 Song, J., Chang, M. Joung, J. *Overview of ITU-T NGN QoS Control*. IEEE Communications Magazine. Septiembre 2007.
- 3 R, Braden, D. Clark, and Shenker. *Integrated Services in the Internet Architecture: An Overview*. IETF RFC 1633, Junio 1994.
- 4 S.Blake *et al.* *An Architecture for Differentiated Services*. IETF RFC 2475. Diciembre 1998.
- 5 Cisco Systems. *Cisco IOS Quality of Service Solutions Configuration Guide*. 2003.
- 6 V. Jacobson, *Congestion Avoidance and Control*, Proc. ACM SIGCOMM '88, 1988, pp. 314–29.
- 7 S.Floyd and V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance*. IEEE/ACM Trans. Networking, vol 1, pp 397-413, Aug 1993.
- 8 Ryu S., Rump C., Qiao C., *Advances in internet congestion control IEEE Communications Surveys & Tutorials*, Vol. 5, no. 1, Third Quarter 2003, pp. 28-39.
- 9 W. Feng *et al.* *A Self-Configuring RED gateway*. Proc. INFOCOM 99. 1999.
- 10 J. Aweya, M. Ouellette, and D. Y. Montuno, *A Control Theoretic Approach to Active Queue Management*, *Computer Networks*, vol. 36, no. 2–3, 2001, pp. 203–35.
- 11 T. J. Ott, T. V. Lakshman, and L. Wong, *SRED: Stabilized RED*, Proc. IEEE INFOCOM '99, 1999, <ftp://ftp.telcordia.com/pub/tjo/SRED.ps>
- 12 W. Feng *et al.*, *Blue: A New Class of Active Queue Management Algorithms*. Technical Report CSE-TR-387-99, University of Michigan, 1999.
- 13 S. Kunniyur and R. Srikant, *Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management*, Technical Report, UIUC, Feb. 2001.
- 14 P.G. Kulkarni, S.I. McClean, G.P. Parr, M.M. Black. *Lightweight ProActive Queue Managemnet*. IEEE Transactions on Network and Service Management. Vol 3. No.2 Second Quarter 2006.
- 15 Cisco Systems. *Implementing Cisco Quality of Service v2.1*. 2004.
- 16 Vali, Dimitra – Paskalis, Sarantis – Kloxyllos, Alexandros. *A survey of Internet QoS Signaling*. IEEE Communications Surveys & Tutorials. Fourth Quarter 2004.
- 17 Ramabhadran, S. Pasquale, J. *The Stratified Round Robin Scheduler: Design, Analysis and Implementation*. IEEE/ACM Transactions on Networking- Vol 14. No.6. December 2006.
- 18 A. Parekh and R. Gallager, *A generalized processor sharing approach to flow control in integrated services networks: the single-node case*. IEEE/ACM



- Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.
- 19 S. Aidarous, T. Plevyak, P. Levine, *Managing IP networks: challenges and opportunities*. Volumen 7 de IEEE Press Series on Network Management, Series Editors Series. Wiley-IEEE, 2003
  - 20 L. Zhang Virtual Clock: A new Traffic control algorithm for packet switching networks. In Proc. ACM SIGCOMM '90, Philadelphia, PA. Sept 1990, pp 19-19.
  - 21 A. Parekh, and R. Gallager. A generalized processor sharing approach to flow control –The single node case. In Proc. INFOCOM '92. 1992.
  - 22 S. Golestani. A self-clocked fair queueing scheme for broadband applications. In Proc. IEEE INFOCOM '94. Toronto, CA, June 1994. pp 636-646.
  - 23 J.C.R Bennett and H. Zhang. *WF<sup>2</sup>Q Worst-case fair weighted fair queueing*. July 95. Submitted to INFOCOM 96.
  - 24 G. Chuanxiong. *SRR, an O(1) time complexity packet scheduler for flows in multi-service packet networks*. In Proc ACM SIGCOMM 2001, San Diego, CA, Aug. 2001 pp 211-222.
  - 25 L. Lenzini, E. Mingozzi and G. Stea. *Aliquem: a novel DRR implementation to achieve better latency and fairness at O(1) complexity*. In Proc 10<sup>th</sup> Int. Workshop on Quality of Service (IWQoS2002), pp.77-88.
  - 26 M. Scheedhar and G. Varghese. *Efficient fair queueing using deficit round-robin*. In Proc. ACM SIGCOMM '95 pp231-242.
  - 27 S. Jha. M. Hassan. *Engineering Internet QoS*. Artech House 2002.
  - 28 S. Alvarez , *QoS for IP/MPLS Networks*. Cisco Press. 2006.
  - 29 M.H. McGregor and W. Shi: *Deficits for Bursty Latency-critical Flows:DRR++*. Proc. Of ICON'00, Singapore, Sept 5-8, 2000.
  - 30 *Understanding and Configuring MDRR/WRED on the Cisco 12000 Series Internet Router*. Doc ID 18841, [http://www.cisco.com/warp/public/63/mdrr\\_wred\\_overview.html](http://www.cisco.com/warp/public/63/mdrr_wred_overview.html).
  - 31 V. Ananthakrishnan ,B. Kelly. *Operation of Modified Deficit Round Robin in M-Series Routers*, Juniper Networks, [http://www.juniper.net/solutions/literature/app\\_note](http://www.juniper.net/solutions/literature/app_note)
  - 32 *Huawei Documentation*. Version 07(2008-09-17). SmartAX MA5600/MA5603 Multi-service Access Module Electronic Documentation.
  - 33 *Recomendación ITU-T Y.2001*.
  - 34 *Estudio Integral de Redes de Nueva Generación y Convergencia*. Documento Amarillo. Comisión de Regulación de Telecomunicaciones. Junio 2007.
  - 35 D. Fellows, D. Jones. **DOCSIS Cable Modem Technology**. IEEE Communications Magazine March 2001.
  - 36 *Technical Report DSL Forum TR-059. DSL Evolution – Architecture Requirements for the Support of QoS- Enabled IP Services*. Septiembre 2003.

- 37 X. Xiao, A. Hannan , B. Bailey , L. Ni *Traffici Engineering with MPLS in the Internet*. IEEE Network. March/ April 2000.
- 38 L. Fang , R. Zhang, M. Taylor. *Services – Requirements and Deployment Case Studies*. IEEE Communications Magazine. March 2008.
- 39 D. Stiliadis. *Efficient Fair Queueing Algorithms for Packet-Switched Networks*. IEEE/ACM Transactions on Networking. Vol 6. No.2 April 1998.
- 40 D.Stephens., J. Bennett. H. Zhan. *Implementing Scheduling Algorithms in High-Speed Network*. IEEE Journal on Selected areas in Communications. Vol 17, No.6 June 1999.
- 41 H. Zhang. *Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks*. Proceedings of the IEEE. Vol 83. No.10 October 1995.
- 42 P. Lieshout., M. Mandjes, S. Borst. *GPS Scheduling: Selection of Optimal Weights and Comparison with Strict Priorities*. SIGMetrics/Performance '06 June 26-30,2006. Saint Malo. France.
- 43 H. Chaskar ,. M. Upamanyu. *Fair Scheduling with Tunable Latency: A Round-Robin Approach*. IEEE/ACM Transactions on Network. Vol 11. No.4. August 2003.
- 44 D. Comer, M. Martynov. *Design and Analysis of Hybrid Packet Schedulers*. IEEE INFOCOM 2008 Proceedings.
- 45 L. Lenzini, E. Mingozzi., S. Giovanni. *Bandwidth and Latency Analysis of Modified Deficit Round Robin Scheduling Algorithms*. ACM ValueTools '06 October 11-13 2006.
- 46 Overview of DiffServ for Quality of Service. Cisco Systems 2207.
- 47 T. Onali, L. Atzori. *Traffic Classification and Bandwidth Management in DiffServ-aware Traffic Engineering Architectures*. IEEE ICC 2008 Proceedings.
- 48 S. Wright.,T. Anschutz. *QoS Requirements in DSL Networks*. 0-7803-7975-6/03. 2003.
- 49 J. Aweya, M. Ouellette, and D. Montuno. *An Optimization oriented view of random early detection*. Computer Commun. Col24, pp 1170-1187, 2001.
- 50 S. Floyd, R. Gummadi, and S. Shenker. *Adaptive red: An Algorithm for increasing the robustness of red*. cite-seer.ist.psu.edu/floyd01adaptive.html. 2001.
- 51 J. Evans, C. Filsfil. *Deploying IP and MPLS QoS for Multiservice Network*. Morgan Kaufmann Publishers. 2007.
- 52 B. Caprita, J. Nieth, and W.C. Chan. *Group Round Robin: Improving the fairness and complexity of packet scheduling*. In Proc. Of ACM/IEEE ANCS 2005.
- 53 X. Yuan and Z.Duan. *FRR: a Proportional and worst-case fair round robin scheduler*. In Proc. of IEEE INFOCOM, March 13-17 2005.
- 54 S. Ramabhadram and J. Pasquale. *Stratified round robin: A low complexity packet scheduler with bandwidth fairness and*

*bounded delay*. In Proc. of ACM SIGCOMM. 2003 pp.239-250.

- 55 Cisco Systems. *Quality of Service for Voice over IP*. 2001

### **CURRÍCULO**

#### **Juan Carlos Cuéllar Quiñónez.**

Ingeniero Electricista egresado de la Universidad del Valle, Especialista en Redes y Servicios Telemáticos de la Universidad del Cauca, Especialista en Redes y Comunicaciones de la Universidad Icesi. CCNA (*Cisco Certified*

*Network Associate*) y candidato a optar el título en la Maestría en Telecomunicaciones en la Universidad Pontificia Bolivariana de Medellín. Actualmente está encargado de la jefatura del Departamento de Ciencias Físicas y Tecnológicas y coordina las actividades en el Laboratorio de Redes y Comunicaciones en la Universidad Icesi. Áreas de interés QoS en Redes de Próxima Generación (NGN). 